

# De-randomizing Shannon: The Design and Analysis of a Capacity-Achieving Rateless Code

Hari Balakrishnan, Peter Iannucci, Jonathan Perry, Devavrat Shah

*Department of EECS\**  
*Massachusetts Institute of Technology*  
*Cambridge, MA, USA.*

## Abstract

This paper presents an analysis of spinal codes, a class of rateless codes proposed recently [17]. We prove that spinal codes achieve Shannon capacity for the binary symmetric channel (BSC) and the additive white Gaussian noise (AWGN) channel with an efficient polynomial-time encoder and decoder. They are the first rateless codes with proofs of these properties for BSC and AWGN.

The key idea in the spinal code is the sequential application of a hash function over the message bits. The sequential structure of the code turns out to be crucial for efficient decoding. Moreover, counter to the wisdom of having an expander structure in good codes [21], we show that the spinal code, despite its sequential structure, achieves capacity. The pseudo-randomness provided by a hash function suffices for this purpose.

Our proof introduces a variant of Gallager's result characterizing the error exponent of random codes for any memoryless channel [10, Chapters 5, 7]. We present a novel application of these error-exponent results within the framework of an efficient sequential code. The application of a hash function over the message bits provides a methodical and effective way to de-randomize Shannon's random codebook construction [19].

---

\*HB, PI, and JP are affiliated with CSAIL; DS is affiliated with LIDS.

# 1 Introduction

In a *rateless code*, the codewords (i.e., coded bits or symbols) corresponding to higher-rate encodings are prefixes of lower-rate encodings. Rateless codes have been known since Shannon’s random codebook construction [19], which proved the existence of capacity-achieving codes. Unfortunately, the random codebook is computationally intractable to decode, taking time exponential in the message size. It took several decades of research on coding theory and algorithms before practical rateless codes were discovered for the binary erasure channel (BEC) by Luby (LT codes [14]) and Shokrollahi (Raptor codes [20]). The BEC is a good model for packet losses on the Internet.

For wireless channels, however, packet erasure models give way to more appropriate random bit-flip models (at the link layer) and additive noise models (at the physical layer). Moreover, wireless channel conditions vary with time due to mobility and interference, even over durations as short as a single packet transmission. In this setting, *fixed-rate* (or fixed-length) codes that work well at a fixed (and known) bit-flip probability or signal-to-noise ratio (SNR) are by themselves insufficient to achieve high throughput; they require additional (and complex) heuristics to determine what the channel conditions are, and to pick the right code [2, 3, 13, 25], resulting in a system without any theoretically appealing properties. This task becomes difficult with rapid channel variations, numerous transmission rate alternatives, and multiple transmitters contending for the same wireless channel.

In contrast to fixed-rate codes, a good rateless code will adapt automatically to changing conditions because it will *inherently* transmit just the right amount, whatever the conditions. Because they are a natural fit for time-varying wireless networks, the design of good rateless codes for the *binary symmetric channel* (BSC) and the *additive white Gaussian noise* (AWGN) channel has received renewed interest recently [6, 11, 17]. By “good”, we mean a code that achieves a rate close to channel capacity:  $1 - H(p)$  for the BSC, where  $p$  is the bit-flip probability and  $H(p) = -p \log p - (1-p) \log(1-p)$ , and  $\frac{1}{2} \cdot \log(1 + \text{SNR})$  for the AWGN channel, where SNR is the ratio of the signal power to the noise variance.<sup>1</sup>

In this paper, we prove that a family of rateless codes, called *spinal codes*, achieves capacity over both the BSC and the AWGN channel. Spinal codes are the *first* provably capacity-achieving rateless codes with a polynomial-time encoder and decoder over both these standard channel models. Our work provides for the BSC and AWGN channel what LT [14] and Raptor [20] codes provide for the BEC, but with a rather different approach.

Spinal codes use hash functions satisfying the *pair-wise independence* [15] to produce a sufficiently random codebook. The encoder for a spinal code applies the hash function sequentially over groups of message bits in a structure that resembles a classic convolutional code. The maximum-likelihood (ML) decoder for a spinal code constructs a tree of possibilities by replaying the encoder over various possible input message bits, and computes either the Hamming distance (BSC) or squared Euclidean distance (AWGN) between the received data and the various choices in the tree. A complete tree is, of course, exponential in the message size, but our key result is that one can aggressively prune the decoding tree to obtain an efficient decoder with polynomial computational cost, that still essentially achieves capacity.

Our approach highlights how the SAC property of the hash function provides a way to de-randomize Shannon’s random codebook [19] approach to produce a practical, capacity-achieving rateless code. As such, our proof methods are likely to extend to de-randomize, and possibly render practical, various random coding constructions in Information Theory that have hitherto been widely used to characterize *existential* capacity results (cf. El Gamal and Kim [4]).

---

<sup>1</sup>In this paper  $\log$  means “logarithm to base 2” and  $\ln$  stands for the natural logarithm.

**Prior work.** Raptor codes, though designed primarily for erasure channels (on which they provably achieve capacity), can be extended to AWGN and BSC channels with a belief propagation decoder [16] similar to graphical codes like LDPC [8, 23]. However, not much is known theoretically about how good this code is over these channels. In fact, the capacity of LDPC codes (with an efficient decoder) over both the BSC and AWGN channels, in general, is still unresolved, which is further evidence that the BSC and AWGN channels are non-trivial settings for the design and analysis of good codes.

Recently, an interesting “layered” approach has been developed by Erez, Trott, and Wornell [6] ([11] describes an implementation of this concept) primarily for the AWGN channel, but there is no obvious way to extend it to the BSC. In this approach, a layered rateless code is built upon a capacity-achieving fixed-rate “base” code at the lowest layer. Erez et al. prove that their code achieves capacity over AWGN assuming that the base code achieves capacity at some SNR and the number of layers increases without bound. Our work is an improvement over this layered approach in two ways: first, we resolve an open question they raise about designing an efficient capacity-achieving rateless code for the BSC, and second, it is a more direct and natural construction that does not rely on layering atop a (presumed capacity-achieving) fixed-rate base code.

Structurally, spinal codes are similar to convolutional codes [5, 24], which apply a linear function sequentially over the message bits, but such codes with so-called small state (constraint length) are far from capacity (in large part because of their sequential nature). In contrast, to achieve capacity using linear codes (whether fixed-rate or rateless) over the BEC, prior work suggests that some form of random graph ensemble or expander structure is necessary [8, 21]. Somewhat surprisingly, despite their sequential nature, we are able to establish that spinal codes—by using a hash function with the pairwise independence—achieve capacity.

**Our results.** For the BSC, we show that rateless spinal codes can be encoded in  $O(\frac{n \log n}{\varepsilon^2})$  time and decoded in  $n^{O(1/\varepsilon^3)}$  time, where  $n$  is the number of message bits and  $\varepsilon$  is the *gap to capacity* at which the code is operating (i.e., the achieved rate is within  $\varepsilon$  of capacity). This result holds for  $n = \Omega(1/\varepsilon^5)$ . For the AWGN channel, we establish a similar result with somewhat lower computational cost:  $O(\frac{n \log n}{\varepsilon})$  time for encoding and  $n^{O(1/\varepsilon^2)}$  time for decoding.

Thus, by selecting  $n = \text{poly}(1/\varepsilon)$ , it is possible to operate within  $\varepsilon$  of capacity with an encoding cost  $\text{poly}(1/\varepsilon)$  and decoding cost  $\exp(\text{poly}(1/\varepsilon))$  for both channel models. These costs are comparable to the computational efficiency achieved by the Forney’s concatenation construction [7], as described in Guruswamy’s survey of iterative decoding methods [12] ( $n/\varepsilon^{O(1)}$  for encoding and  $n2^{1/\varepsilon^{O(1)}}$  for decoding). However, the key advantage of spinal codes is that they are rateless, unlike all known good and efficient codes for the BSC; and they are arguably more elegant than the concatenation construction. We have implemented spinal codes in both software and hardware (FPGA) to demonstrate their practicality and high throughput, allowing us to project that a silicon implementation of the design will run at 50 Mbits/s (commercial 802.11b/g speeds) [18]. The experimental results should alleviate concerns about the super-linearity of the encoder and decoder being a barrier to their practical usefulness.

**Method and proof technique.** The key idea is to use the error exponents of random codes as a building block. We apply Gallager’s result characterizing the random coding error exponent for any memoryless channel [10, Chapters 5, 7]. That result, though established for random codes where the codewords for distinct messages are mutually independent, applies even if only pairwise independence between the coded bits holds. The application of this idea to analyze spinal codes is somewhat remarkable because many coded bits of two distinct messages are likely to be highly dependent. The rest of the proof uses probabilistic analysis leveraging the SAC property of the hash function as a de-randomization

strategy, establishing that a sequentially structured code can achieve capacity.

## 2 Overview of Spinal Codes

This section describes the encoder (§2.1) and decoder (§2.2) for spinal codes, which are variants of the methods introduced in [17]. Our discussion here is in the context of the BSC, but the same approach with one addition (direct coding to symbols) works for the AWGN channel, as described in §4.

### 2.1 Encoder

The encoder maps  $n$  input message bits,  $\mathbf{m} = (m_1, \dots, m_n)$  to a stream of coded bits,  $x_1(\mathbf{m}), x_2(\mathbf{m}), \dots$ . These coded bits are transmitted in sequence until the receiver signals that it is done decoding.

**Hash function.** The core of the code is a hash function,  $h$ , which takes two inputs, a  $\nu$ -bit state and  $k$  message bits, and maps them to a new  $\nu$ -bit state. That is,  $h : \{0, 1\}^\nu \times \{0, 1\}^k \rightarrow \{0, 1\}^\nu$ . We choose  $h$  uniformly at random, based on a random seed, from  $\mathcal{H}$ , a family of hash functions with *pair-wise* independence property cf. [15]: each  $x \in \{0, 1\}^\nu$  is mapped uniformly at random (randomness induced by selection of random seed) to any of the  $\{0, 1\}^k$ ; for any  $x \neq x' \in \{0, 1\}^\nu$ ,

$$\mathbb{P}(h(x) = y, h(x') = y') = \mathbb{P}(h(x) = y) \mathbb{P}(h(x') = y') = 2^{-2k}, \quad (1)$$

for any  $y, y' \in \{0, 1\}^k$ .

**Spine.**  $h$  is applied sequentially to  $k$  non-overlapping message bits at a time, producing a sequence of  $\nu$ -bit states called the *spine*. The initial state  $s_0 = 0^\nu$ . Let  $\bar{m}_i = (m_{ki+1} \dots m_{k(i+1)})$  be the  $i^{\text{th}}$   $k$ -bit block of the message  $\mathbf{m}$ . Then, as shown in Figure 1, each successive  $\nu$ -bit value in the spine is generated as

$$s_i = h(s_{i-1}, \bar{m}_{i-1}), \quad 1 \leq i \leq n/k.$$

**Generating coded bits.** The encoder uses the spine values  $s_1, \dots, s_{n/k}$  to produce coded bits in passes. In the first pass, it extracts the most significant bit from each  $\nu$ -bit spine value to produce  $n/k$  coded bits  $x_1, \dots, x_{n/k}$ . In general, in the  $\ell^{\text{th}}$  pass, the encoder extracts the  $\ell^{\text{th}}$  most significant bit of each spine value  $s_1, \dots, s_{n/k}$ , producing coded bits  $x_{(\ell-1)\frac{n}{k}+1}, \dots, x_{\ell\frac{n}{k}}$ . The coding parameters  $k, \nu$  determine various properties of the code. The maximum rate achieved by the code at the end of the  $\ell^{\text{th}}$  pass is  $R_\ell = k/\ell$ ; the lowest achievable rate is  $k/\nu$ .

**Sequential structure of the code.** The combination of the encoder's iterative structure and the SAC property of the hash function gives the code a unique balance. On the one hand, two messages that differ by one or more bits will have very different codewords, allowing analysis using random coding techniques. On the other hand, this divergence in the output is structured in such a way as to allow an efficient decoder.

In a spinal code, the output bits  $x_i, x_{i+\frac{n}{k}}, x_{i+2\frac{n}{k}}, \dots$  are fully determined by the first  $i \cdot k$  bits of the message  $\mathbf{m}$ . Two messages that first differ in the  $i^{\text{th}}$  block of  $k$  bits have the same first  $i-1$  spine values, and have statistically independent subsequent spine values (i.e., the later values are “very different”).

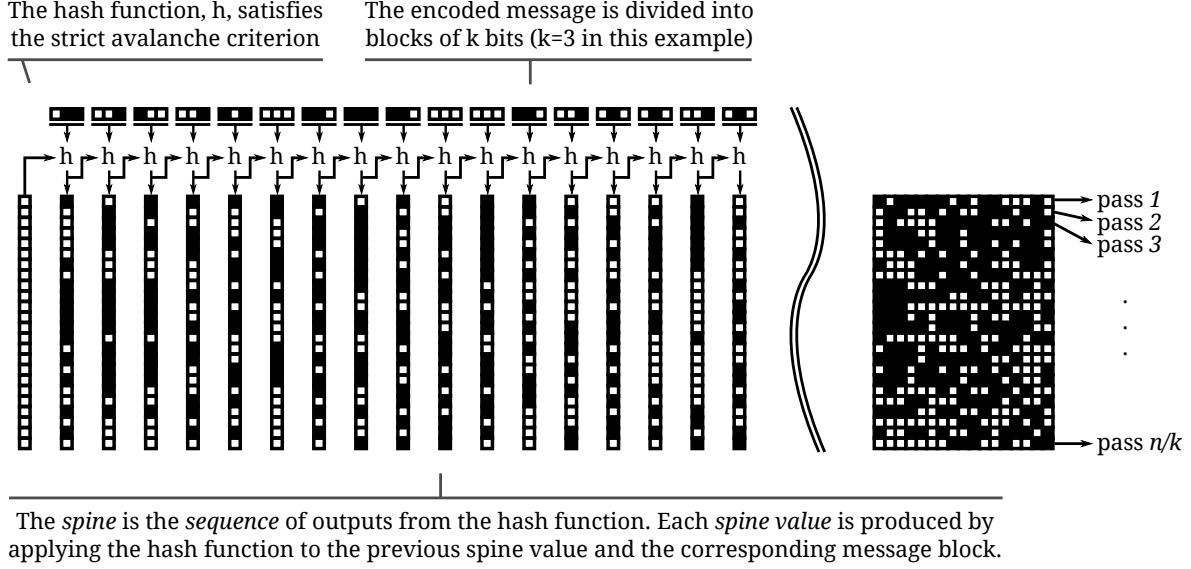


Figure 1: Encoder for the BSC. Each dark square is a “1”, each white square is a “0”.

## 2.2 Decoder

**Decoding over a tree.** Maximum likelihood (ML) decoding over the BSC boils down to a search for the encoded message whose Hamming distance is nearest to the received message. Because the spinal encoder applies the hash function sequentially, input messages with a common prefix will also have a common spine prefix. The key to exploiting this structure is to decompose the total distance into a sum over spine values. If we break the received bits  $\mathbf{y}$  into sub-vectors  $\mathbf{y}_1, \dots, \mathbf{y}_{n/k}$  containing symbols from spine values  $s_1, \dots, s_{n/k}$  of the correct message, and similarly if we break  $\mathbf{x}(\mathbf{m}')$  for the candidate message  $\mathbf{m}'$  into  $n/k$  vectors of bits  $\mathbf{x}_1(s'_1), \dots, \mathbf{x}_{n/k}(s'_{n/k})$  that depend on spine values  $s'_1, \dots, s'_{n/k}$  (corresponding to message  $\mathbf{m}'$ ), then the cost function decomposes as

$$d_H(\mathbf{y}, \mathbf{x}(\mathbf{m}')) = \sum_{i=1}^{n/k} d_H(\mathbf{y}_i, \mathbf{x}_i(s'_i)). \quad (2)$$

A summand  $d_H(\mathbf{y}_i, \mathbf{x}_i(s_i))$  only needs to be computed once for all messages that share the same spine value  $s_i$ . The following algorithm takes advantage of this property.

Ignoring hash function collisions (as established in the proof of Theorem 1, this happens with very low probability), decoding can be recast as a search over a tree of message prefixes. The root of this *decoding tree* is  $s_0$ , and corresponds to the zero-length message. Each node at depth  $d$  corresponds to a prefix of length  $kd$  bits, and is labeled with the final spine value  $s_d$  of that prefix. Every node has  $2^k$  children, connected by edges  $e = (s_d, s_{d+1})$  representing a choice of  $k$  message bits  $\bar{m}_e$ . As in the encoder,  $s_{d+1} = h(s_d, \bar{m}_e)$ . By walking back up the tree to the root and reading  $k$  bits from each edge, we can find the message prefix for a given node.

To the edge incident on node  $s_d$ , we assign a *branch cost*  $d_H(\mathbf{y}_d, \mathbf{x}_d(s_d))$ . Summing the branch costs on the path from the root to a node gives the *path cost* of that node, equivalent to the sum in Eq.(2). The ML decoder finds the leaf with the lowest cost, and returns the corresponding complete message. The sender continues to send successive passes until the receiver signals that the message has been decoded correctly. The receiver stores all the symbols it receives until the message is decoded correctly.

**Pruning the tree.** Decoding along the tree has exponential complexity. A natural greedy approximation is to prune the tree by maintaining a small number of candidates with the lowest path costs at each depth, while exploring the tree from root to leaves. Iteratively, at each depth, expand the retained (up to)  $B$  candidates into  $B2^k$  possible candidates at the next depth of the tree. Compute the path cost of all of these  $B2^k$  candidates and retain the  $B$  out of them with the lowest possible path cost (break ties arbitrarily). We use the term *beam width* to refer to the parameter  $B$ , as this tree exploration and pruning method is called beam search [22] in AI, and known as the  $M$ -algorithm [1] in the coding literature, where it has been proposed for decoding convolutional codes. We show the somewhat surprising and noteworthy result that this simple greedy method essentially achieves channel capacity when used for spinal decoding.

**Encoding and decoding complexity.** The encoder produces  $n/k$  spine values each with  $\nu$  bits. Since the cost of producing  $\nu$  hash bits from a  $\nu$ -bit and  $k$ -bit input is  $O(\nu + k)$ , the encoding cost (due to hash function calculations) scales as  $O((\nu + k)n/k) = O(n(1 + \nu/k))$ . The decoder uses the pruned tree search over  $n/k$  depth tree with each depth requiring sorting  $B \cdot 2^k$  numbers as well as  $B \cdot 2^k$  hash operations. Therefore, the total decoding cost scales as  $O(nB2^k(k + \log B + \nu))$ .

### 3 Performance of Spinal Codes over the BSC

The principal result of this section is a proof of Theorem 1 (stated below), showing the polynomial-time encoder and greedy tree-pruning decoder for spinal codes achieve Shannon capacity over the BSC.

**Model: Memoryless Channel in Discrete Time.** A noisy channel is described by an input alphabet  $\mathcal{I}$ , an output alphabet  $\mathcal{O}$ , and a collection of probability measures  $\mathcal{P} = (P_i, i \in \mathcal{I})$ , defined over  $\mathcal{O}$ : when input  $i \in \mathcal{I}$  is transmitted over the channel, the received output is distributed over  $\mathcal{O}$  according to  $P_i$ . The communication channel is memoryless: the output of the channel at any time depends only on the input at that time, independent of past transmissions. That is, when  $x_1, \dots, x_T$  are transmitted on the channel, the probability (density) that the output is  $y_1, \dots, y_T$  is  $\prod_{t=1}^T P_{x_t}(y_t)$ .

The BSC is memoryless. In a BSC with bit-flip probability  $p \in (0, 1/2)$ ,  $\mathcal{I} = \mathcal{O} = \{0, 1\}$ ,  $P_0(0) = P_1(1) = 1 - p$ , and  $P_0(1) = P_1(0) = p$ .

**Theorem 1.** *Consider an  $n$ -bit message encoded with a spinal code with  $k \geq 1$  and  $\nu = \Theta(k^2 \log n)$  operating over a BSC with parameter  $p \in (0, 1/2)$ . Then, the greedy decoder with  $B = n^{O(k^3)}$  decodes all but the last  $O(k^3 \log n)$  message bits successfully with probability at least  $1 - 1/n^2$ , achieving a rate*

$$R \geq C - O\left(\frac{C^2}{k}\right), \quad \text{where } C = 1 - H(p). \quad (3)$$

The randomness in Theorem 1 is induced by the channel conditions and the code construction. For  $n = \omega(k^3 \log n)$ , the theorem says that essentially all bits are decoded (to decode *all* the bits, we can append  $O(k^3 \log n)$  “tail” bits to the end of each input message). For  $n \geq k^5$ , the loss of rate due to these tail bits is  $O((k^4 \log n)/n) = o(1/k)$ . Therefore, the code achieves a rate within  $O(1/k)$  of the capacity of the BSC, making it a good rateless code. The encoder complexity scales as  $O(nk \log n)$ ; the decoder complexity scales as  $n^{O(k^3)}$ .

**Proof plan.** The rest of this section establishes this result with the following plan. We start by recalling Gallager's result on the probability of error for a random code, which requires the codewords associated with distinct messages to be completely independent. We present a useful variant of this result, which requires only *pairwise* independence (a property, we show to be satisfied by different enough messages under application of the hash function, see Proposition 5). We then discuss a corollary of the result for a code operating at a rate close to the capacity, and establish that spinal codes can operate at a rate near the capacity (so that the corollary will apply). Finally, we use these propositions to prove Theorem 1 in two stages: first, assuming no hash function collisions, and then showing that the collision probability is small.

### 3.1 Error probability of random codes

The random code for a message of  $n$  bits is constructed using a distribution  $Q$  over the input symbols. For the BSC, the input symbols are  $\{0, 1\}$  and a capacity-achieving random code utilizes  $Q$  such that  $Q(0) = Q(1) = 1/2$ . The code maps an  $n$ -bit message,  $\mathbf{m} \in \{0, 1\}^n$ , to a  $T$ -symbol codeword  $\mathbf{x}(\mathbf{m}) = (x_1(\mathbf{m}), \dots, x_T(\mathbf{m}))$  by drawing each of the  $x_t(\mathbf{m})$ ,  $1 \leq t \leq T$  independently at random according to  $Q$ . In the random code, introduced by Shannon and considered by Gallager, all  $\mathbf{x}(\mathbf{m})$  are independent across  $\mathbf{m} \in \{0, 1\}^n$ . We consider a random code with *pairwise* independence across messages.

**Property 1** (*Pairwise independent random code for the BSC*). *A code that maps every  $n$ -bit message  $\mathbf{m} \in \{0, 1\}^n$  to a random codeword of  $T$  bits,  $\mathbf{x}(\mathbf{m})$ , so that (i) for a given  $\mathbf{m}$ ,  $x_1(\mathbf{m}), \dots, x_T(\mathbf{m})$  are i.i.d. and uniformly distributed over  $\{0, 1\}$ , (ii) for any  $\mathbf{m} \neq \mathbf{m}'$ ,  $\mathbf{x}(\mathbf{m})$  and  $\mathbf{x}(\mathbf{m}')$  are independent of each other, and (iii) the joint distribution of all codewords is symmetric.*

For pairwise independent random codes, the following variant of Gallager's error-exponent result [9] [10, Theorem 5.6.1, Example 1] holds (proof in Appendix C):

**Lemma 2.** *Consider a BSC with parameter  $p \in (0, 1/2)$  and capacity  $C = 1 - H(p)$ . Given a pairwise independent random code for the BSC of message length  $n$ , code length  $T$ , and rate  $R = n/T < C$ , let the decoder operate using the Maximum Likelihood (ML) rule to produce an estimate  $\hat{\mathbf{m}}$  when message  $\mathbf{m} \in \{0, 1\}^n$  is transmitted. Then the probability of decoding error,  $P_e = 2^{-n} \left( \sum_{\mathbf{m} \in \{0, 1\}^n} \mathbb{P}(\mathbf{m} \neq \hat{\mathbf{m}}) \right)$ , for  $R = 1 - H(q)$  with  $p < q$  satisfies:*

- (a)  $P_e \leq 2^{-TD(q||p)}$ , where  $D(q||p) = q \log \frac{q}{p} + (1 - q) \log \frac{1-q}{1-p}$ , if  $q \leq \sqrt{p}/(\sqrt{p} + \sqrt{1-p})$ , and
- (b)  $P_e \leq 2^{-T(1-R-2\log(\sqrt{p}+\sqrt{1-p}))}$ , otherwise.

### 3.2 Error probability at rates $R$ close to capacity $C$

**Lemma 3.** *Consider the same setup as Lemma 2 with rate  $R = n/T = 1 - H(q)$  close to capacity  $C = 1 - H(p)$ , that is,  $q \approx p$ . Then,*

$$P_e \leq 2^{-TD(q||p)} \approx 2^{-T\kappa_p(C-R)^2}, \quad (4)$$

where  $\kappa_p^{-1} = \Theta(p(1-p)(\log \frac{1-p}{p})^2)$ .

*Proof.* From Lemma 2, for all  $q$  close enough to  $p$ ,  $P_e \leq 2^{-TD(q||p)}$ . Now, consider  $p$  fixed and let  $F(q) = D(q||p)$  be function of  $q$ . Then, by Taylor's expansion of  $F(q)$  around  $p$ ,

$$F(q) = F(p) + F'(p)(q - p) + F''(\theta)(q - p)^2/2, \quad (5)$$

for  $\theta \in [p, q]$ . Noting that  $F'(x) = \log\left(\frac{x(1-p)}{(1-x)p}\right)$  and  $F''(x) = \frac{1}{x(1-x)\ln 2}$ , we see that  $F(p) = F'(p) = 0$ , and that for  $q \approx p$ ,

$$F(q) \approx \frac{(q-p)^2}{p(1-p)\ln 4}. \quad (6)$$

For the entropy function  $H(x) = -x \log x - (1-x) \log(1-x)$ , using the first-order Taylor expansion, we obtain that for  $q \approx p$ ,

$$H(q) \approx H(p) + \log\left(\frac{1-p}{p}\right)(q-p). \quad (7)$$

Since  $R = 1 - H(q)$  and  $C = 1 - H(p)$ ,

$$q - p \approx \frac{(C - R)}{\log\left(\frac{1-p}{p}\right)}. \quad (8)$$

The desired claim follows from (6) and (8).  $\square$

### 3.3 Rates achievable by spinal codes

The following claim shows that a spinal code over the BSC can achieve rates arbitrarily close to the channel capacity,  $C$ , for large  $k$ . Hence, Lemma 3 is applicable.

**Claim 4.** *There exists  $L \geq 1$  so that the rate induced by the spinal code at the end of pass  $L$  satisfies*

$$C - R = \Theta\left(\frac{C^2}{k}\right).$$

*Proof.* Consider  $L$  such that  $\frac{k}{L-2} \geq C > \frac{k}{L-1}$ . These conditions may be rewritten as

$$\frac{k}{C} + 1 < L \leq \frac{k}{C} + 2 \quad \frac{C}{L} < C - \frac{k}{L} \leq \frac{2C}{L}.$$

Hence,  $L = \Theta\left(\frac{k}{C}\right)$ , and  $C - R = \Theta\left(\frac{C}{L}\right)$ . Together  $C - R = \Theta\left(\frac{C^2}{k}\right)$ .  $\square$

### 3.4 Proof of Theorem 1

We now establish that by the end of pass  $L$ , chosen as above, decoding happens with high probability. We shall prove that if  $B = n^{O(k^3)}$ , with high probability, for  $i^* = \Theta(k^2 C^{-4} \kappa_p^{-1} \log n)$ , when processing the  $i^{\text{th}}$  spine value, all non-pruned codewords either agree with the  $i - i^*$  true spine values (so there are less than  $B$  of them), or are less likely than the true spine (so cannot cause the true spine to be pruned out). As a consequence, the true spine is never pruned, so the decoder manages to decode all but  $i^* k = \Theta(k^3 C^{-4} \kappa_p^{-1} \log n)$  bits.

The following proposition is an implication of the strong avalanche criterion.

**Proposition 5.** *Let  $\mathbf{m}, \mathbf{m}'$  be two messages differing in message block  $\bar{m}_i$ . Let  $\{s_j\}$  and  $\{s'_j\}$  be the spines for  $\mathbf{m}$  and  $\mathbf{m}'$ , respectively. Then,*

$$\mathbb{P}(\exists j \in \{1, \dots, g\} : s_{i+j} = s'_{i+j}) \leq g \cdot 2^{-\nu}$$

*If such a  $j$  does not exist, then all the bits of  $s_{i+1} \dots s_{i+g}$  are independent of bits of  $s'_{i+1} \dots s'_{i+g}$ , and each of them has a uniform independent distribution.*



*Proof.* Due to the pairwise independence property of hash function, when two different inputs are passed through the hash function, the output bits corresponding to these input bits are independent of each other and each of them is distributed independently and uniformly. Therefore, the chance of two different inputs producing the same output is  $2^{-\nu}$ . By the union bound, the probability of such an event happening over a series of  $g$  spine values is bounded by  $g \cdot 2^{-\nu}$ . By iteratively applying the property that when spine values differ at some stage  $t$ , the bits produced at stage  $t + 1$  are independent and uniformly distributed, we conclude that if all spines are different, their bits are independent and distributed uniformly.  $\square$

**Proving Theorem 1 assuming no collisions.** We establish Theorem 1 assuming no hash function collisions. Later we show that collisions happen with low probability. We require  $\nu = \Theta(k^2 \log n)$  with a large-enough constant multiplier in  $\Theta(\cdot)$  term. Throughout, we will assume that this  $\mathbf{m} \in \{0, 1\}^n$  is a fixed choice that was transmitted. Establishing that with high probability (with respect to all randomness in code construction and channel noise) it gets decoded will imply all messages get decoded with high probability due to symmetry of the random-code and memoryless property of the BSC noise model (or more generally, any memoryless channel).

**Lemma 6.** *Consider the greedy spinal decoder operating after all coded bits of the  $L$  passes are received. Assuming no hash collisions, the decoder decodes all but the last  $O(k^3 \log n)$  bits correctly with probability  $1 - O(1/n^4)$ .*

*Proof.* Consider message  $\mathbf{m}$  that was transmitted and any other message  $\mathbf{m}'$  that differs from  $\mathbf{m}$  in any of the first  $k$  bits. In the absence of hash collisions, as per Lemma 5, codewords of  $\mathbf{m}$  and  $\mathbf{m}'$  are independent of each other and each of their bits is independent and uniformly distributed over  $\{0, 1\}$ . That is,  $\mathbf{m}$  and  $\mathbf{m}'$  satisfy Property 1.

If we restrict our attention to codewords generated from the first  $i$  spine values, that is, codewords of length  $N = iL$ , there are  $2^{ik-k}$  codewords, one each for a message  $\mathbf{m}'$  that differs from  $\mathbf{m}$  in any of the first  $k$  bits. As established above, the pair  $\mathbf{m}$  and any other  $\mathbf{m}'$  satisfies Property 1. Using Lemmas 2 and 3, we obtain that the probability that any of the  $2^{ik-k}$  messages (that differ from  $\mathbf{m}$  in any of the first  $k$  bits) is more likely than the original message  $\mathbf{m}$  is bounded above by  $P_e(i)$ , where (with  $k$  large enough for Lemma 3 to be applicable)

$$P_e(i) = 2^{-N\kappa_p(C-R)^2} = 2^{-iL \frac{\kappa_p C^4}{k^2}}. \quad (9)$$

That is, for  $i^* = \Theta(k^2 C^{-4} \kappa_p^{-1} \log n)$ , the probability of such an error is bounded above by  $1 - 1/n^6$  (with a suitably large constant factor in the  $\Theta(\cdot)$  term for  $i^*$ ). Therefore, after processing the first  $i^*$  spines, the only messages that can have a higher likelihood than the original message are those that do not differ from  $\mathbf{m}$  in the first  $k$  bits. There are at most  $2^{i^*k-k}$  such messages and hence if  $B = 2^{i^*k} = n^{O(k^3)}$ , then the original message will not be pruned out.

Now we apply the above argument inductively. Consider a stage  $j$  where the only messages that are not pruned out and have likelihood higher than the original message  $\mathbf{m}$  are those that differ from  $\mathbf{m}$  in a bit position between  $jk - i^*k + 1$  to  $jk$ . Now when the decoder moves to stage  $j + 1$ , messages that are not pruned out are expanded by factor  $2^k$ . Among these, consider the messages that start differing from the original message in any of the  $k$  bit positions:  $jk - i^*k + 1, \dots, jk - i^*k + k$ . By applying the same argument as we did above, it follows that at the end of stage  $j + 1$ , all of these  $2^{i^*k-k}$  messages will have likelihood smaller than the original message with probability at least  $1 - 1/n^6$ .

The above invariant together with the union bound implies that at the end of stage  $n/k$ , the original message is preserved in the  $B$  candidates with probability at least  $1 - O(1/n^5)$ . Further, the most likely

$2^{i^*k}$  of these  $B$  candidates are those that have correct  $n - i^*k$  prefix bits. That is, the decoder manages to decode all but last  $O(i^*k) = O(k^3 \log n)$  bits correctly.  $\square$

**Dealing with collisions.** The above proof uses the fact that in the absence of collisions, given the original message  $\mathbf{m}$  of interest and any other message  $\mathbf{m}'$  that differs from  $\mathbf{m}$  in the first  $k$  bits, their corresponding codewords  $\mathbf{x} = \mathbf{x}(\mathbf{m})$  and  $\mathbf{x}' = \mathbf{x}'(\mathbf{m}')$  satisfy Property 1. Therefore, the probability of any such message  $\mathbf{m}'$  having likelihood higher than  $\mathbf{m}$  is at most  $O(1/n^6)$  as desired. Note that this is precisely the argument that is used inductively along with the union bound to establish the claim. Therefore, it is sufficient to establish that the effect of collision is negligible for this step only.

We wish to show that the effect of collisions is small, using the following plan. As stated in Lemma 7, we will identify an event  $\mathcal{E}$  so that conditioned on it happening, Property 1 is satisfied as above; and, the probability of event  $\mathcal{E}^c$  is  $O(1/n^6)$ . Using this, we will establish that the probability of any such message  $\mathbf{m}'$  having likelihood higher than  $\mathbf{m}$  continues to remain at most  $O(1/n^6)$  as desired.

**Lemma 7.** *Let  $\mathbf{m}$  be the  $i^*k$  prefix bits of an uncoded message. Consider any other message prefix  $\mathbf{m}'$  of the same length, with any of the first  $k$  bits differing from  $\mathbf{m}$ . Then there exists event  $\mathcal{E}$  so that*

- (a) *Conditioned on event  $\mathcal{E}$ , all pairs of messages  $(\mathbf{m}, \mathbf{m}')$ , satisfy Property 1.*
- (b) *The probability of  $\mathcal{E}^c$  is  $O(1/n^6)$ .*

The proof of Lemma 7 is in Appendix A. Using the above propositions, we complete the proof of Theorem 1 here. Define the event **err** as the one in which the likelihood of an undesirable message (prefix)  $\mathbf{m}'$  is higher than original message (prefix)  $\mathbf{m}$ . Conditioned on event  $\mathcal{E}$ , as per Lemma 7(a), Property 1 is satisfied by all relevant codeword pairs as desired in the proof of Theorem 1 in the absence of collisions. Therefore, conditioned on event  $\mathcal{E}$ , and the arguments presented earlier for the no-collision case, it follows that  $\mathbb{P}(\text{err}|\mathcal{E}) = O(1/n^6)$ . That, together with Lemma 7(b), yields

$$\mathbb{P}(\text{err}) = \mathbb{P}(\text{err} \cap \mathcal{E}) + \mathbb{P}(\text{err} \cap \mathcal{E}^c) \leq \mathbb{P}(\text{err}|\mathcal{E})\mathbb{P}(\mathcal{E}) + \mathbb{P}(\mathcal{E}^c) \leq \mathbb{P}(\text{err}|\mathcal{E}) + O(1/n^6) = O(1/n^6).$$

This completes the proof of Theorem 1.

## 4 Performance of Spinal Codes over AWGN

The main result of this section is that spinal codes achieve Shannon capacity over the AWGN channel with a polynomial-time encoder and decoder. The arguments are similar to the BSC case.

**AWGN channel model.** The transmitter's primary resource is power, measured as the squared value of the output symbols. Typically, for regulatory and practical reasons, the average power should be  $\leq P$  for some  $P$ . If an  $n$ -bit message  $\mathbf{m} = (m_1, \dots, m_n)$  is mapped to  $T$  symbols  $\mathbf{x}(\mathbf{m}) = (x_1(\mathbf{m}), \dots, x_T(\mathbf{m}))$ , then the power of  $\mathbf{x}(\mathbf{m})$  is  $\frac{1}{T} \sum_i x_i^2(\mathbf{m})$ . The rate of such a code is  $R = n/T$  bits/symbol. When these symbols are transmitted over the AWGN channel, the receiver sees  $\mathbf{y} = \mathbf{x} + \mathbf{z}$ , where noise-vector  $\mathbf{z} = (z_1, \dots, z_T)$  has i.i.d. Gaussian components with mean 0 and variance  $\sigma^2$ . The capacity of this channel is  $C_{\text{awgn}}(P) = \frac{1}{2} \log_2(1 + \text{SNR})$  bits/symbol, where  $\text{SNR} = \frac{P}{\sigma^2}$  denotes the signal-to-noise ratio.

**Encoder and Decoder.** The procedure described in §2.1 for generating output symbols for the BSC is modified slightly to produce a stream of coded symbols in  $\mathbb{R}$ . The modified encoder generates coded symbols from each  $\nu$ -bit spine value: in the first pass, the encoder produces  $n/k$  symbols  $x_1, \dots, x_{n/k}$

using the  $c$  most significant bits of  $s_1, \dots, s_{n/k}$ , respectively. In the next pass, the next  $c$  most-significant bits are used, and so on.

The sequence of  $c$  input bits is treated as a binary number  $b \in \{0, \dots, 2^c - 1\}$ . The encoder computes each output symbol as  $x_i = \Phi^{-1}(\gamma + (1 - 2\gamma)u)\sqrt{P}$ , where  $\Phi$  is the CDF of standard Gaussian,  $u = (b + 1/2)/2^c$ , and  $\gamma = \Phi(-\beta)$ . The symbols generated are in the range  $[-\beta\sqrt{P}, \beta\sqrt{P}]$ , and within that range they are distributed like a Gaussian with mean 0 and variance  $P$ , quantized into  $2^c$  equally-probable values. When  $\beta, c \rightarrow \infty$ , the coded symbols will be i.i.d. Gaussian.

The only change to the decoder is to use the squared  $\ell_2$  Euclidean distance instead of the Hamming distance in (2). The intuition is that in each case, given the channel parameters, the distance metric gives (up to normalization) the log likelihood that a message is correct given the observation  $\mathbf{y}$ .

**Performance over AWGN.** The following result shows that spinal codes achieve nearly optimal rates over the AWGN channel in a rateless manner with efficient encoding and decoding algorithms.

**Theorem 2.** *Consider an AWGN channel with noise variance bounded below by  $\sigma_{\min}^2$ . Consider a spinal code constrained to have average power  $\leq P$ , with  $k > \frac{1}{2} \log(1 + P/\sigma_{\min}^2)$ . Let the code map  $n$  message bits to coded symbols with  $\beta, c$  as per (33), with  $\varepsilon = 1/k$  and  $\sigma_{\min}$  in place of  $\sigma$  in (33). Let  $\nu = \Theta(k^2 c \log n)$ , and let the decoder operate with  $B = n^{O(k^2)}$ . Then the decoder will correctly decode all but the last  $O(k^2 \log n)$  message bits with probability at least  $1 - 1/n^2$  within time  $T$  such that the induced rate  $R = n/T$  satisfies*

$$R \geq C_{\text{awgn}}(P) - O(1/k). \quad (10)$$

The proof involves a choice of parameters  $\beta = \Theta(\sqrt{\log k})$  and  $c = \Theta(|\log \text{SNR}| + |\log \sigma_{\min}| + \log k)$ . These parameters depend on  $\sigma_{\min}$ , to bound the “dynamic range” of the channel capacity.

*Proof.* The highest rate at which the code can operate is  $k$ . Choose  $k$  large enough so that  $k > \frac{1}{2} \log(1 + \text{SNR}_{\max})$ , where  $\text{SNR}_{\max} = P/\sigma_{\min}^2$ . Now let  $\varepsilon = 1/k$  and assign the remaining parameters as in Claim 11 (in Appendix B; mirrors Lemma 3).

The proof of Claim 4 still holds with  $C = C_{\text{awgn}} - 1/k$ , so a rate  $R$  such that  $C - R = \Theta(\frac{C^2}{k})$  is achievable. That is,  $R = C_{\text{awgn}} - \Theta(1/k)$ . Theorem 2 proceeds according to the same arguments as the proof of Theorem 1, with Claim 11 replacing Lemma 3, to achieve (for large enough  $k$ ) the bound

$$P_e \leq 2^{-N(C_{\text{awgn}} - 1/k - R)} = 2^{-\Theta(NC^2/k)}. \quad (11)$$

Subsequently, (9) is replaced by

$$P_e(i) = 2^{-\Theta(iLC^2/k)}. \quad (12)$$

That is,  $i^*$  is chosen to be  $\Theta(kC^{-2} \log n)$  rather than  $\Theta(k^2 C^{-4} \kappa_p^{-1} \log n)$ , and now  $B = 2^{i^*k} = n^{O(k^2)}$ , rather than  $n^{O(k^3)}$ . Finally,  $\nu$  is required to be  $\Theta(k \log n)$  rather than  $\Theta(k^2 \log n)$ .  $\square$

## 5 Conclusion

We proved that spinal codes achieve Shannon capacity for the BSC and AWGN channels with an efficient polynomial-time encoder and decoder; they are the first rateless codes with these properties. The key idea in the spinal code is the application of a hash function in a sequential manner over the message bits. The sequential structure of the code turns out to be crucial for efficient decoding, while

the pair-wise independence of the hash function provides enough pseudo-randomness to ensure that the code essentially achieves capacity.

The key idea in the proof is an unusual application of a variant of Gallager’s famous result characterizing the error exponent of random codes for any memoryless channel; the use of this result is unconventional because the spinal code is not a traditional random code. Our work provides a methodical and effective way to de-randomize Shannon’s random codebook construction, and as such, applies immediately to all discrete memoryless channels and will likely generalize to other random coding arguments in Information Theory.

## References

- [1] J. Anderson and S. Mohan. Sequential coding algorithms: A survey and cost analysis. *IEEE Trans. on Comm.*, 32(2):169–176, 1984.
- [2] John Bicket. Bit-Rate Selection in Wireless Networks. Master’s thesis, Massachusetts Institute of Technology, February 2005.
- [3] J. Camp and E. Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-Layer Implementation and Experimental Evaluation. In *Proc. of the ACM MobiCom Conf.*, pages 315–326, San Francisco, CA, September 2008.
- [4] A. El-Gamal and Y.-H. Kim. *Network Information Theory*. Cambridge University Press, 2012.
- [5] P. Elias. Coding for two noisy channels. In *Third London Symposium on Information Theory*, pages 61–76, 1955.
- [6] Erez, U. and Trott, M. and Wornell, G. Rateless Coding for Gaussian Channels. *IEEE Trans. Inform. Theory*, 58(2):530–547, 2012.
- [7] G. D. Forney. *Concatenated Codes*. MIT Press, 1966.
- [8] R. Gallager. Low-density parity-check codes. *IRE Trans. Information Theory*, 8(1):21–28, 1962.
- [9] R. Gallager. A simple derivation of the coding theorem and some applications. *Information Theory, IEEE Transactions on*, 11(1):3–18, 1965.
- [10] R.G. Gallager. *Information theory and reliable communication*. Wiley, 1968.
- [11] A. Gudipati and S. Katti. Strider: Automatic rate adaptation and collision handling. In *SIGCOMM*, 2011.
- [12] V. Guruswamy. Iterative Decoding of Low-Density Parity Check Codes. *Bull. of the European Association for Theoretical Computer Science (EATCS)*, 90, September 2006.
- [13] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *MobiSys*, June 2008.
- [14] M. Luby. LT codes. In *FOCS*, 2003.
- [15] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge Univ Pr, 2005.

- [16] R. Palanki and J.S. Yedidia. Rateless codes on noisy channels. In *ISIT*, 2005.
- [17] Jonathan Perry, Hari Balakrishnan, and Devavrat Shah. Rateless spinal codes. In *HotNets-X*, October 2011.
- [18] Jonathan Perry, Peter Iannucci, Kermin Elliott Fleming, Hari Balakrishnan, and Devavrat Shah. A rateless wireless communication system using spinal codes. In *Preprint; available on request*, January 2012.
- [19] C.E. Shannon. Communication in the presence of noise. *Proc. of the IRE*, 37(1):10–21, 1949.
- [20] A. Shokrollahi. Raptor codes. *IEEE Trans. Info. Theory*, 52(6), 2006.
- [21] M. Sipser and D.A. Spielman. Expander codes. *Information Theory, IEEE Transactions on*, 42(6):1710–1722, 1996.
- [22] V. Steinbiss, B.H. Tran, and H. Ney. Improvements in beam search. In *2nd Intl. Conf. on Spoken Language Processing*, 1994.
- [23] A.I. Vila Casado, M. Griot, and R.D. Wesel. Informed dynamic scheduling for Belief-Propagation decoding of LDPC codes. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 932–937, 2007.
- [24] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.
- [25] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *SIGCOMM*, 2009.
- [26] A. Webster and S. Tavares. On the design of s-boxes. In *Advances in Cryptology-CRYPTO'85 Proceedings*, pages 523–534, 1986.

## A Proof of Lemma 7

*Proof.* To construct event  $\mathcal{E}$ , consider the original message (prefix)  $\mathbf{m}$  and any other message (prefix)  $\mathbf{m}'$  that differs from  $\mathbf{m}$  in any of the first  $k$  bits. Both these messages are of length  $i^*k$ . Given  $\mathbf{m}$ , denote all such message (prefixes)  $\mathbf{m}'$  as  $\mathcal{M}'(\mathbf{m})$  (note that  $|\mathcal{M}'(\mathbf{m})| = 2^{i^*k-k}$ ).

At the end of  $L$  passes, the codewords generated based on these (prefix) messages are of length  $N = i^*L$ . Let them be  $\mathbf{x}$  and  $\mathbf{x}'$  respectively. We wish to evaluate the joint probability of  $\mathbf{x} = \mathbf{b}$  and  $\mathbf{x} = \mathbf{b}'$  for any  $\mathbf{b}, \mathbf{b}' \in \{0, 1\}^N$  (effectively, we are assuming a re-indexing of the coded bits so that the first  $L$  coded bits depend on the first spine value, the next  $L$  coded bits depends on the next spine value, and so on). Since the messages  $\mathbf{m}$  and  $\mathbf{m}'$  differ in the first  $k$  bits, by the property of the hash function (Proposition 5), the  $\nu$  bits of the first spine values for the messages are i.i.d. uniform random bits. If the first spine values of the two messages differ (i.e., no collision), which happens with probability  $1 - 2^{-\nu}$ , the  $\nu$  bits of the second spine values for the two messages are i.i.d. uniform random bits, and so on. Let  $E_j$  be the event that the first  $j$  spine values for both messages are not the same (i.e., no collision amongst first  $j$  spine values). Then  $\mathbb{P}(E_j|E_{j-1}) = 1 - 2^{-\nu}$ . Therefore, for  $j \geq 1$  and since  $E_j \subset E_{j-1}$ ,

$$\begin{aligned}
 \mathbb{P}(E_j) &= \mathbb{P}(E_j \cap E_{j-1}) = \mathbb{P}(E_j|E_{j-1})\mathbb{P}(E_{j-1}) \\
 &= (1 - 2^{-\nu})\mathbb{P}(E_{j-1}) \\
 &= (1 - 2^{-\nu})^j.
 \end{aligned} \tag{13}$$

Now, conditioned on  $E_{j-1}$ , the  $L$  coded bits generated from the  $j^{\text{th}}$  spine value in  $\mathbf{x}$  and  $\mathbf{x}'$  are i.i.d. and uniformly distributed. Therefore, (with notation  $\mathbf{x}_{i,j} = (x_i, \dots, x_j)$ , etc.)

$$\begin{aligned}
\mathbb{P}(\mathbf{x} = \mathbf{b}, \mathbf{x}' = \mathbf{b}') &\geq \mathbb{P}(\mathbf{x} = \mathbf{b}, \mathbf{x}' = \mathbf{b}', E_1) \\
&= \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1) \mathbb{P}(\mathbf{x}_{1,L} = \mathbf{b}_{1,L}, \mathbf{x}'_{1,L} = \mathbf{b}'_{1,L}, E_1) \\
&\geq \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1) \left( \mathbb{P}(\mathbf{x}_{1,L} = \mathbf{b}_{1,L}, \mathbf{x}'_{1,L} = \mathbf{b}'_{1,L}) - \mathbb{P}(E_1^c) \right) \\
&= \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1) \left( \mathbb{P}(\mathbf{x}_{1,L} = \mathbf{b}_{1,L}) \mathbb{P}(\mathbf{x}'_{1,L} = \mathbf{b}'_{1,L}) - \mathbb{P}(E_1^c) \right) \\
&= \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1) \left( 2^{-2L} - 2^{-\nu} \right) \\
&= \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1) 2^{-2L} (1 - 2^{-\nu+2L}) \\
&= (1 - 2^{-\nu+2L}) \mathbb{P}(\mathbf{x}_{1,L} = \mathbf{b}_{1,L}) \mathbb{P}(\mathbf{x}'_{1,L} = \mathbf{b}'_{1,L}) \mathbb{P}(\mathbf{x}_{L+1,N} = \mathbf{b}_{L+1,N}, \mathbf{x}'_{L+1,N} = \mathbf{b}'_{L+1,N} | E_1).
\end{aligned} \tag{14}$$

Here, we have used the fact that the distribution of  $\mathbf{x}_{L+1,N}, \mathbf{x}'_{L+1,N}$  is conditionally independent of  $\mathbf{x}_{1,L}, \mathbf{x}'_{1,L}$  given  $E_1$ . (14) then sets up a recursion, leading to the following:

$$\mathbb{P}(\mathbf{x} = \mathbf{b}, \mathbf{x}' = \mathbf{b}') \geq \mathbb{P}(\mathbf{x} = \mathbf{b}) \mathbb{P}(\mathbf{x}' = \mathbf{b}') (1 - 2^{-\nu+2L})^{i^*}. \tag{15}$$

Given this, with respect to the underlying probability space,  $\Omega$ , we can define an event  $\mathcal{E}_{\mathbf{m}, \mathbf{m}'}$  with

$$\mathbb{P}(\mathcal{E}_{\mathbf{m}, \mathbf{m}'}) = (1 - 2^{-\nu+2L})^{i^*}, \tag{16}$$

so that we have

$$\mathbf{1}_{\{\mathbf{x}=\mathbf{b}, \mathbf{x}'=\mathbf{b}'\}} = \mathbb{P}(\mathbf{x} = \mathbf{b}) \mathbb{P}(\mathbf{x}' = \mathbf{b}') \mathbf{1}_{\{\mathcal{E}_{\mathbf{m}, \mathbf{m}'}\}} + q(\mathbf{b}, \mathbf{b}') \mathbf{1}_{\{\mathcal{E}_{\mathbf{m}, \mathbf{m}'}^c\}}, \tag{17}$$

where  $\mathbf{1}_{\{E\}}(\cdot)$  is the indicator random variable of event  $E$  with  $\mathbf{1}_{\{E\}}(\omega) = 1$  if  $\omega \in E$  and 0 otherwise for  $\omega \in \Omega$  and  $q(\cdot, \cdot)$  represents the conditional probability distribution of  $\mathbf{x}, \mathbf{x}'$  given  $\mathcal{E}^c$ . Equivalently, what we have is an event  $\mathcal{E}_{\mathbf{m}, \mathbf{m}'}$  with property (16) such that

$$\mathbb{P}(\mathbf{x} = \mathbf{b}, \mathbf{x}' = \mathbf{b}' | \mathcal{E}_{\mathbf{m}, \mathbf{m}'}) = \mathbb{P}(\mathbf{x} = \mathbf{b}) \mathbb{P}(\mathbf{x}' = \mathbf{b}'). \tag{18}$$

Now define

$$\mathcal{E} = \bigcap_{\mathbf{m}' \in \mathcal{M}'(\mathbf{m})} \mathcal{E}_{\mathbf{m}, \mathbf{m}'}. \tag{19}$$

Since  $\mathcal{E} \subset \mathcal{E}_{\mathbf{m}, \mathbf{m}'}$  for any  $\mathbf{m}' \in \mathcal{M}'(\mathbf{m})$  and from (17), the conditional distribution of  $\mathbf{x}, \mathbf{x}'$  with respect to  $\mathcal{E}_{\mathbf{m}, \mathbf{m}'}$  is uniform, it follows that, for any  $\mathbf{m}' \in \mathcal{M}'(\mathbf{m})$ ,

$$\mathbb{P}(\mathbf{x} = \mathbf{b}, \mathbf{x}' = \mathbf{b}' | \mathcal{E}) = \mathbb{P}(\mathbf{x} = \mathbf{b}) \mathbb{P}(\mathbf{x}' = \mathbf{b}'). \tag{20}$$

Finally, by (16) and union bound, it follows that

$$\mathbb{P}(\mathcal{E}^c) = O(i^* 2^{N+2L-\nu}) \tag{21}$$

Therefore, choosing

$$\nu = (N + 2L + \log i^*) + 6 \log n = \Theta(k^2 \log n), \tag{22}$$

with an appropriately large constant leads to

$$\mathbb{P}(\mathcal{E}^c) = O(1/n^6), \tag{23}$$

as desired, completing the proof of Lemma 7.  $\square$

## B Error probability of random codes over AWGN

As in §3.1, we consider random codes with only pairwise independent codewords across messages.

**Property 8** (*Pairwise independent random code for AWGN with distribution  $Q$* ). A code that maps every  $n$  bit message  $\mathbf{m} \in \{0,1\}^n$  to a random codeword of  $T$  real numbers,  $\mathbf{x}(\mathbf{m})$ , so that (i) for a given  $\mathbf{m}$ ,  $x_1(\mathbf{m}), \dots, x_T(\mathbf{m})$  are i.i.d. with distribution  $Q$ , (ii) for any  $\mathbf{m} \neq \mathbf{m}'$ ,  $\mathbf{x}(\mathbf{m})$  and  $\mathbf{x}(\mathbf{m}')$  are independent of each other, and (iii) the joint distribution of all codewords is symmetric.

This definition allows us to state the following variant of Gallager's error-exponent result [10, Theorem 7.3.2] for a random code on the AWGN channel. The coded symbols have distribution  $Q$  over a finite set  $\Omega \subset \mathbb{R}$ .

**Lemma 9.** Consider an AWGN channel with noise variance  $\sigma^2$  and a pairwise independent random code for AWGN with distribution  $Q$ , message length  $n$ , code length  $T$ , and rate  $R = n/T < C$ . Then the probability of error under ML decoding is bounded by

$$P_e \leq 2^{-T(E_o(Q)-R)}, \quad \text{where} \quad E_o(Q) = -\log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\mathbb{R}} \left[ \sum_{j \in \Omega} Q(j) \exp \left( -\frac{(y-j)^2}{4\sigma^2} \right) \right]^2 dy \right\}. \quad (24)$$

Next, we want to specialize this bound for the spinal code symbol distribution described in §4. Given  $c$ ,  $\beta$  and  $P$ , let

$$\Omega = \left\{ \Phi^{-1}(\gamma + (1 - 2\gamma)u)\sqrt{P} : \gamma = \Phi(-\beta), u = \frac{b + 1/2}{2^c}, b \in \{0, \dots, 2^c - 1\} \right\}.$$

where  $\Phi$  is the CDF of the standard Gaussian. By construction,  $|\Omega| = 2^c$  and  $\Omega \subset [-\beta\sqrt{P}, \beta\sqrt{P}]$ . The distribution over  $b$  is uniform, as in the case of the spinal encoder, and hence each  $j \in \Omega$  is equally likely with probability  $2^{-c}$ . This leads to the following result.

**Lemma 10.** For the channel and code of Lemma 9 with uniform distribution over  $\Omega$  (with parameters  $\beta, c, P$ ), the probability of error under ML decoding is bounded above as

$$P_e \leq 2^{-T(E'-R)}, \quad \text{where} \quad E' = \max_{\zeta > 1} \left( \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2(1 + 1/\zeta^2)} \right) - \frac{2r(\beta)}{\ln 2} - \frac{(2\zeta + 1)\Delta^2 \log e}{4\sigma^2} \right). \quad (25)$$

*Proof.* Lemma 9 indicates that a random code generated with this distribution would have error probability

$$P_e \leq 2^{-T(E-R)}, \quad \text{where} \quad E = -\log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\mathbb{R}} \left[ \sum_{j \in \Omega} 2^{-c} \exp \left( -\frac{(y-j)^2}{4\sigma^2} \right) \right]^2 dy \right\}. \quad (26)$$

The expression above is explicit but opaque. We can simplify it by rewriting the summation over discrete  $j \in \Omega$  as an integral over  $x \in [-\beta\sqrt{P}, \beta\sqrt{P}]$  with Gaussian density, provided that we construct a suitable function  $\delta(x)$  so that  $j = x + \delta(x)$  is distributed according to  $\Omega$ . Extracting  $\delta(x)$  from the resulting integrand will yield a tractable expression.

Using the mean value theorem and the properties of the Gaussian, the separation between two adjacent elements in  $\Omega$  can be bounded above by

$$\Delta \equiv \Delta(\beta, c, P) = \frac{\beta\sqrt{P} \exp(\beta^2/2)}{2^{c-1}}.$$

Now consider the following thought experiment. First, sample a Gaussian variable with mean 0 and variance  $P$ . If the outcome is within  $[-\beta\sqrt{P}, \beta\sqrt{P}]$ , map it to a nearby value in  $\Omega$  so that the induced distribution over elements of  $\Omega$  is uniform (equiprobable quantization); if the outcome is not within  $[-\beta\sqrt{P}, \beta\sqrt{P}]$ , reject it (truncation). The rejection probability,  $r(\beta)$ , is  $2(1 - \Phi(\beta)) = O(\frac{1}{\beta} \exp(-\beta^2/2))$ . We can relate the quantized value  $j$  to the sampled Gaussian value  $x$  by an additive discretization error  $\delta(x) = j - x$ . From these properties, it follows that the discrete summation involving probabilities  $2^{-c}$  over  $\Omega$  in (26) can be replaced by a Riemann integral over the Gaussian density with mean 0 and variance  $P$ , normalized by  $1/(1 - r(\beta))$ , and limited to the range  $[-\beta\sqrt{P}, \beta\sqrt{P}]$ :

$$E = -\log \left\{ \frac{1}{\sqrt{8\pi^3 P^2 \sigma^2}} \int_{\mathbb{R}} \left[ \int_{-\beta\sqrt{P}}^{\beta\sqrt{P}} (1 - r(\beta))^{-1} \exp\left(-\frac{x^2}{2P}\right) \exp\left(-\frac{(y-x-\delta(x))^2}{4\sigma^2}\right) dx \right]^2 dy \right\}, \quad (27)$$

By construction,  $|\delta(x)| \leq \Delta$ . We can pull  $\delta(x)$  out of the integrand by placing a multiplicative bound on  $\exp(-(y-x-\delta(x))^2/4\sigma^2)$  in terms of  $\exp(-(y-x)^2/4\sigma^2)$  and a small error term involving  $\Delta$ . Let  $\zeta > 1$  be a large constant. Then if  $|y-x| \leq \zeta|\delta(x)|$ ,

$$\exp\left(-\frac{(y-x-\delta(x))^2}{4\sigma^2}\right) \leq \exp\left(-\frac{(y-x)^2}{4\sigma^2}\right) \exp\left(\frac{(2\zeta+1)\Delta^2}{4\sigma^2}\right). \quad (28)$$

Otherwise,  $|y-x| \geq \zeta|\delta(x)|$  and hence

$$\exp\left(-\frac{(y-x-\delta(x))^2}{4\sigma^2}\right) \leq \exp\left(-\frac{(y-x)^2}{4\sigma^2(1+1/\zeta)^2}\right). \quad (29)$$

From (27)-(29), it follows that (using approximation  $\log(1-x) \approx -x/\ln 2$  and treating  $r(\beta)$  small or equivalently  $\beta$  large),

$$E \geq -\frac{2r(\beta)}{\ln 2} - \frac{(2\zeta+1)\Delta^2 \log e}{4\sigma^2} - \log \left\{ \frac{1}{\sqrt{8\pi^3 P^2 \sigma^2}} \int_{\mathbb{R}} \left[ \int_{\mathbb{R}} \exp\left(-\frac{x^2}{2P}\right) \exp\left(-\frac{(y-x)^2}{4\sigma^2(1+1/\zeta)^2}\right) dx \right]^2 dy \right\}. \quad (30)$$

As established in [10, Eq (7.4.21)],

$$\begin{aligned} & -\log \left\{ \frac{1}{\sqrt{8\pi^3 P^2 \sigma^2 (1+1/\zeta)^2}} \int_{\mathbb{R}} \left[ \int_{\mathbb{R}} \exp\left(-\frac{x^2}{2P}\right) \exp\left(-\frac{(y-x)^2}{4\sigma^2(1+1/\zeta)^2}\right) dx \right]^2 dy \right\} \\ &= \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2(1+1/\zeta)^2} \right). \end{aligned} \quad (31)$$

Combining (30) and (31), we obtain the desired result.  $\square$

**Claim 11.** *With an appropriate choice of parameters, for a pairwise independent random code over AWGN with distribution  $Q$ , the probability of error for a rate  $R < C_{\text{awgn}} - \varepsilon$  is bounded as*

$$P_e \leq 2^{-T(C_{\text{awgn}} - \varepsilon - R)} \quad (32)$$

*Proof.* For a given small enough  $\varepsilon > 0$ , select

$$\begin{aligned} \zeta &= 9\text{SNR}/\varepsilon, \quad \text{with } \text{SNR} = P/\sigma^2, \\ \beta &\text{ large enough so that } r(\beta) = \frac{\varepsilon \ln 2}{6} \quad \text{where recall } r(\beta) = 2(1 - \Phi(\beta)), \\ c &\text{ large enough so that } \Delta = \frac{2\varepsilon\sigma^2}{9\sqrt{P}} \quad \text{where recall } \Delta = \frac{\beta\sqrt{P} \exp(\beta^2/2)}{2^{c-1}}. \end{aligned} \quad (33)$$



This selection leads to  $\beta = \Theta(\sqrt{\log 1/\varepsilon})$  and  $c = \Theta(\log \text{SNR}_{\max} + \log \sigma_{\min} + \log 1/\varepsilon)$  with  $\text{SNR}_{\max} = P/\sigma_{\min}^2$ . Now, with these choices of parameters and using the fact that  $\frac{1}{2} \log(1+x)$  is a 1-Lipschitz function, we obtain from (25) that

$$E' \geq C_{\text{awgn}} - \varepsilon. \quad (34)$$

□

## C Variation of Gallager's result: Pairwise independent random code and discrete memoryless channel

Here we present a derivation of a variation of Gallager's result about the error exponent (or error probability) for a random code under the ML decoding rule for any discrete memoryless channel. The variation assumes the pairwise independence property of random codewords rather than complete independence. Effectively, we observe that the proof technique of Gallager [9, 10] requires only pairwise independence. Since results identical to Lemmas 2 and 9 were derived by specializing them for the BSC and AWGN channel respectively (see [10, Chapters 5, 7]), the justification of these two Lemmas follow.

**Pairwise independent random code.** Consider  $n$ -bit messages in  $\{0,1\}^n$ . Let  $Q$  be distribution over  $\mathcal{J}$ . Under a pairwise random code, using  $Q$ , of rate  $R = n/T$ , each message  $\mathbf{m} \in \{0,1\}^n$  is mapped to a random codeword  $\mathbf{x}(\mathbf{m}) \in \mathcal{J}^T$  such that

- (a) For any  $\mathbf{m} \in \{0,1\}^n$  and  $\mathbf{i} = (i_1, \dots, i_T) \in \mathcal{J}^T$ ,

$$\mathbb{P}(\mathbf{x}(\mathbf{m}) = \mathbf{i}) = \prod_{t=1}^T Q(i_t). \quad (35)$$

- (b) For any  $\mathbf{m} \neq \mathbf{m}' \in \{0,1\}^n$  and  $\mathbf{i}, \mathbf{i}' \in \mathcal{J}^T$ ,

$$\mathbb{P}(\mathbf{x}(\mathbf{m}) = \mathbf{i}, \mathbf{x}(\mathbf{m}') = \mathbf{i}') = \mathbb{P}(\mathbf{x}(\mathbf{m}) = \mathbf{i}) \times \mathbb{P}(\mathbf{x}(\mathbf{m}') = \mathbf{i}') \quad (36)$$

- (c) The joint distribution of all codewords is symmetric.

**Maximum likelihood decoding.** To transmit message  $\mathbf{m}$ , the codeword  $\mathbf{x}(\mathbf{m})$  is sent over the channel, producing output  $\mathbf{y}$ . The ML rule produces an estimate  $\hat{\mathbf{m}}$  so that

$$\mathbb{P}(\mathbf{y}|\mathbf{x}(\hat{\mathbf{m}})) = \max_{\mathbf{m}' \in \{0,1\}^n} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}')). \quad (37)$$

A decoding error occurs if  $\hat{\mathbf{m}} \neq \mathbf{m}$ .

**Probability of error.** Let  $P_{e\mathbf{m}}$  denote the probability of decoding error when  $\mathbf{m}$  was transmitted.  $P_{e\mathbf{m}}$  is average of probability of error over all randomly chosen codes. As before, the overall probability of error is

$$P_e = \frac{1}{2^n} \left( \sum_{\mathbf{m} \in \{0,1\}^n} P_{e\mathbf{m}} \right). \quad (38)$$

Due to symmetry in the random code,  $P_{e\mathbf{m}}$ , the average probability of error over all choices of codes, is the same for all  $\mathbf{m}$ . Therefore,  $P_e$  equals  $P_{e\mathbf{m}}$  for any given  $\mathbf{m}$ .

**Theorem 3.** *Given the above setup, for any  $\mathbf{m} \in \{0, 1\}^n$ ,*

$$P_{e\mathbf{m}} \leq 2^{-T(-\rho R + E_o(\rho, Q))}, \quad (39)$$

for any  $0 < \rho \leq 1$  with

$$E_o(\rho, Q) = -\log \left[ \sum_{j \in \mathcal{O}} \left( \sum_{i \in \mathcal{I}} Q(i) P_{ij}^{\frac{1}{1+\rho}} \right)^{1+\rho} \right]. \quad (40)$$

The best bound is achieved by optimizing for choice of  $\rho, Q$ . Specifically, define

$$E_o(R) = \max_{\rho, Q} \left[ -\rho R + E_o(\rho, Q) \right]. \quad (41)$$

Then, Theorem 3 implies the bound  $P_{e\mathbf{m}} \leq 2^{-NE_o(R)}$ . This bound when specialized to the BSC and AWGN channel (with proper choice of  $\rho, Q$  in (41)) results in Lemmas 2 and 9 (see [10, Chapters 5, 7] for details).

*Proof of Theorem 3.* The proof is essentially identical to that in [9], presented here for completeness. Consider a message  $\mathbf{m} \in \{0, 1\}^n$ . Then,

$$\mathbb{P}(\mathbf{m} \neq \hat{\mathbf{m}}) = \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m})) \phi_{\mathbf{m}}(\mathbf{y}), \quad (42)$$

where

$$\phi_{\mathbf{m}}(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m})) \leq \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}')) \text{ for some } \mathbf{m}' \neq \mathbf{m}, \\ 0 & \text{otherwise.} \end{cases} \quad (43)$$

This can be upper bounded as

$$\phi_{\mathbf{m}}(\mathbf{y}) \leq \left[ \frac{\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}}{\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}}} \right]^\rho, \quad \rho > 0. \quad (44)$$

From (42) and (44), we obtain

$$\mathbb{P}(\mathbf{m} \neq \hat{\mathbf{m}}) \leq \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left[ \sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}} \right]^\rho, \quad \rho > 0. \quad (45)$$

Now recalling that it is a pairwise independent random code and averaging both sides with respect to

this random code, we obtain for  $0 < \rho \leq 1$ ,

$$\begin{aligned}
P_{em} &\equiv \mathbb{E}[\mathbb{P}(\mathbf{m} \neq \hat{\mathbf{m}})] \\
&\leq \mathbb{E}\left[\sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left[\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right]^\rho\right] \\
&= \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{E}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left[\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right]^\rho\right] \\
&= \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{E}_{\mathbf{x}(\mathbf{m})}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \mathbb{E}\left[\left[\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right]^\rho \middle| \mathbf{x}(\mathbf{m})\right]\right] \\
&\stackrel{(a)}{\leq} \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{E}_{\mathbf{x}(\mathbf{m})}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left(\mathbb{E}\left[\left[\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right] \middle| \mathbf{x}(\mathbf{m})\right]\right)^\rho\right] \\
&= \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{E}_{\mathbf{x}(\mathbf{m})}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left(\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{E}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}} \middle| \mathbf{x}(\mathbf{m})\right]\right)^\rho\right] \\
&\stackrel{(b)}{=} \sum_{\mathbf{y} \in \mathcal{O}^T} \mathbb{E}_{\mathbf{x}(\mathbf{m})}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}))^{\frac{1}{1+\rho}} \left(\sum_{\mathbf{m}' \neq \mathbf{m}} \mathbb{E}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right]\right)^\rho\right]. \tag{46}
\end{aligned}$$

Here, we use the notation  $\mathbb{E}_{\mathbf{x}(\mathbf{m})}$  to explicitly note that the randomness is with respect to  $\mathbf{x}(\mathbf{m})$ ; (a) follows from Jensen's inequality for conditional expectation and fact that  $f(x) = x^\rho$  is a concave function for  $0 < \rho \leq 1$ ; and (b) follows from the pairwise independence of  $\mathbf{x}(\mathbf{m})$  and  $\mathbf{x}(\mathbf{m}')$  for any pair of messages  $\mathbf{m} \neq \mathbf{m}'$ . Now due to symmetry of the random coding distribution, it follows that  $\mathbb{E}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right]$  is the same for all  $\mathbf{m}'$  (including  $\mathbf{m}$ ) and equals

$$\mathbb{E}\left[\mathbb{P}(\mathbf{y}|\mathbf{x}(\mathbf{m}'))^{\frac{1}{1+\rho}}\right] = \sum_{\mathbf{i} \in \mathcal{I}^T} Q^T(\mathbf{i}) \mathbb{P}(\mathbf{y}|\mathbf{i})^{\frac{1}{1+\rho}}, \tag{47}$$

where  $Q^T(\mathbf{i}) = \prod_{t=1}^T Q(i_t)$ . Therefore, from (46) and the fact that  $n = RT$ , we have

$$P_{em} \leq 2^{\rho RT} \sum_{\mathbf{y} \in \mathcal{O}^T} \left[\sum_{\mathbf{i} \in \mathcal{I}^T} Q^T(\mathbf{i}) \mathbb{P}(\mathbf{y}|\mathbf{i})^{\frac{1}{1+\rho}}\right]^{1+\rho}. \tag{48}$$

Now using the property of memoryless channels and random codes, we have that

$$Q^T(\mathbf{i}) \mathbb{P}(\mathbf{y}|\mathbf{i})^{\frac{1}{1+\rho}} = \prod_{t=1}^T Q(i_t) \mathbb{P}(y_t|i_t)^{\frac{1}{1+\rho}}. \tag{49}$$

Using this product-form in (48) and exchanging sums and products, we have

$$\begin{aligned}
P_{em} &\leq 2^{\rho RT} \prod_{t=1}^T \sum_{y_t \in \mathcal{O}} \left[\sum_{i_t \in \mathcal{I}} Q(i_t) \mathbb{P}(y_t|i_t)^{\frac{1}{1+\rho}}\right]^{1+\rho} \\
&\stackrel{(a)}{=} 2^{\rho RT} \left[\sum_{j \in \mathcal{O}} \left(\sum_{i \in \mathcal{I}} Q(i) P_{ij}^{\frac{1}{1+\rho}}\right)^{1+\rho}\right] \\
&\stackrel{(b)}{=} 2^{\rho RT} 2^{-TE_o(\rho, Q)} \\
&= 2^{-T(-\rho R + E_o(\rho, Q))}. \tag{50}
\end{aligned}$$

Here, (a) uses the definitions of random code and memoryless channel, and (b) follows from the definition of  $E_o(\rho, Q)$ .  $\square$